Automated Map Reading: Image Based Localisation in 2-D Maps Using Binary Semantic Descriptors

Pilailuck Panphattarasap and Andrew Calway

Abstract— **We describe a novel approach to image based localisation in urban environments which uses semantic matching between images and a 2-D cartographic map. This contrasts with the majority of existing approaches which use image to image database matching. We use highly compact binary descriptors to represent locations, indicating the presence or not of semantic features, which significantly increases scalability and has the potential for greater invariance to variable imaging conditions. The approach is also more akin to human map reading, making it better suited to human-system interaction. In this initial study we use semantic features relating to buildings and road junctions in discrete viewing directions. CNN classifiers are used to detect the features in images and we match descriptor estimates with location tagged descriptors derived from the 2-D map to give localisation. The descriptors are not sufficiently discriminative on their own, but when concatenated sequentially along a route, their combination becomes highly distinctive and allows localisation even when using non-perfect classifiers. Performance is further improved by taking into account left or right turns over a route. Experimental results obtained using Google StreetView and OpenStreetMap data show that the approach has considerable potential, achieving localisation accuracy of around 85% using routes corresponding to approximately 200 meters.**

I. INTRODUCTION

Large scale image based localisation and place recognition have been looked at extensively as an alternative to infrastructure dependent sensing such as GPS, especially when operating in urban environments. The majority of approaches adopt *image to image database matching*, in which environment images are matched to a database of location tagged images [1]. These have demonstrated impressive performance, but they are limited in three key respects. The first is scalability - localisation is dependent on having a very large database of images or image features and thus scaling to very large areas is problematic. The second relates to invariance - matching is impacted significantly by variable imaging conditions and so maintaining performance at all times over extended periods is challenging. Finally, such schemes do not align well with how it is believed that humans perceive and undertake location-based activities, which are thought to be based on some form of 2-D map representation [2], [3], [4]. Thus these approaches do not lend themselves naturally to human-system interaction.

Motivated by the above, we adopt an alternative approach using *image to 2-D map matching*, in which we link images to semantic features on a 2-D cartographic map to give

The authors are with Department of Computer Science, University of Bristol, UK {pp12907,andrew.calway}@bristol.ac.uk

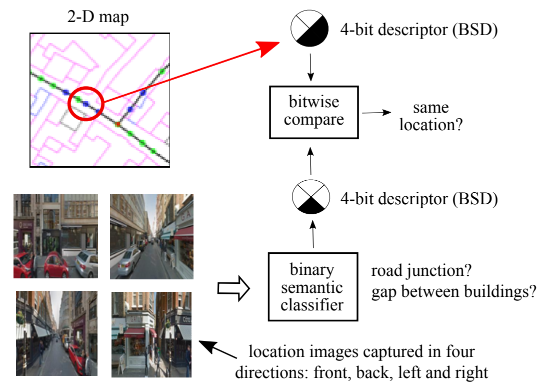


Fig. 1. Binary semantic descriptors (BSDs). 4-bit binary descriptors are used to represent locations indicating the presence or not of semantic features in 4 directions (front/back facing - junctions; left/right facing - gaps between buildings). These are derived from a 2-D map and compared bitwise with descriptors estimated via classifiers from images captured in the same directions to establish localisation w.r.t the map. On their own the descriptors are not sufficiently distinctive, but when combined sequentially along routes as shown in Figure 2, then localisation becomes possible.

localisation. This is akin to human map reading, in which visual appearance is matched with semantic information perceived on a map, such as buildings, road layout, etc. This makes the approach potentially better suited to human-system interaction. Moreover, the abstraction and compression provided by semantic description gives potential for significant gains in scalability - our semantic descriptors are many orders of magnitude smaller than images or sets of image features - and improved invariance to variable imaging conditions, since via training, semantic feature detection can be made less dependent on specific instances.

We present preliminary investigations into the approach. Our central idea is to characterise locations by a small number of semantic features which can be extracted from 2-D maps such as road junctions, buildings, etc, and represent each location by a binary semantic descriptor (BSD), with each bit indicating the presence or not of a feature in a given viewing direction. This gives a very compact representation (we use 4-bit descriptors in this work) and so increases scalability. We design classifiers to recognise the features in images, allowing us to estimate the descriptors and hence in principle recognise locations by matching with location tagged descriptors derived from the 2-D map. The approach is illustrated in Figure 1.

However, BSDs are not sufficiently discriminative on their own; many locations have the same descriptors and thus localisation is not possible. To overcome this, we concatenate

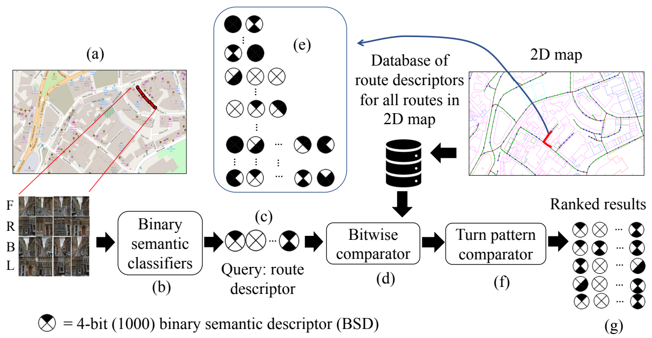


Fig. 2. Route based localisation. (a) Images captured in four directions (front, back, left and right facing) at locations along a route are converted to BSDs using binary classifiers (b) and concatenated to produce route descriptors (c). These are compared bit-wise (d) with a database of ground-truth BSDs (e) derived from the 2-D map to determine the closest matching route. Routes are then compared in terms of their turn patterns (f) to give a final ranking of possible locations of the images w.r.t the 2-D map (g).

descriptors sequentially along routes, which results in highly distinctive *route descriptors*, so that localisation becomes possible even when using non-perfect classifiers. In essence, the *pattern of semantic features* along a route become unique providing the route is sufficiently long (we achieved localisation after approximately 200 meters). Moreover, when travel direction between locations is also taken into account, e.g. left and right turns, performance is further improved. This routes based approach is illustrated in Figure 2. Note that it is feasible because of the compact representation, i.e. a small number of bits per location, something that would be difficult to achieve using the large feature-based representations used in image to image database matching.

We present an implementation using Google StreetView (GSV) and OpenStreetMap[1] (OSM) data, with the latter providing vector maps and the former giving 360 degree images at regularly spaced locations along roads. We used road junctions and gaps between buildings as our semantic features, assuming the former to be present or not in front and back facing views, and the latter to be present or not in left and right facing views. This gives a 4-bit BSD for each location. We trained convolutional neural network (CNN) classifiers to recognise the features in images, achieving accuracy of around 75%. In experiments on an area of around 2.5 km^2, we achieved localisation accuracy in excess of 85% when using routes consisting of 20 or more locations, corresponding to distances of approximately 200 meters. Although initial localisation is delayed as the route evolves, once bootstrapped to the correct location, the method successfully tracks the route at the same rate as location images are captured and achieves this using a significantly smaller database than required in image to image database matching.

II. RELATED WORK

In image to image database matching, environments are represented by location tagged images or image features [1]. Key concerns are invariance to changes in viewpoint and to appearance caused by different lighting and weather conditions. For example, the FAB-MAP algorithms [5] use image features with a degree of viewpoint invariance to give large-scale matching over routes of up to 1000 km, whilst other methods seek to deal with changing appearance through invariant representations [6], storing multiple representations [7], learning models of appearance change [8] or more recently via deep learning methods [9], [10]. However, in all cases, large scale localisation requires large scale storage, in the order of hundreds of gigabytes [5].

Work has been done on using semantic information for localisation. For example, in [11], [12], semantically labelled landmarks and objects extracted from images are used to represent locations for place recognition, and in [13], localisation based on particle filtering is implemented using object detection in images coupled with bespoke semantically labelled maps based on cars and building windows, for example. These approaches demonstrate the potential of semantic features to provide invariance and reduced representation size, but are not linked to cartographic map data and thus do not naturally scale to very large area localisation.

Others have used 2-D map data to aid the estimation of 6-D camera pose using a combination of GPS and images [14], [15], [16], [17]. Building edges and planar facades extracted from images are used to align with 2-D and 2.5-D maps, geo-localised using GPS, and give improved estimates of camera position and orientation. However, the focus is on obtaining precise metric pose estimates for applications such Augmented Reality [16] and there is a reliance on having clear views of building facades, making these approaches difficult to extend to general localisation.

The closest work to that presented here is that described in [19] and [20], both in the context vehicle localisation. A CNN approach is used in [19] to recognise semantic features in images, such as junctions, number of lanes, bike lanes, one-way vs two-way, etc. The network training is based on labels obtained from OSM and images from GSV, as in our approach. The classifier outputs are used to validate GPS map locations for self-driving car applications, rather than for general localisation, and locations are considered in isolation, in contrast to our use of route information. The latter is used in a number of localisation techniques based on map matching, see e.g. [21], [22], in which visual odometry based on feature matching is used to generate route trajectories, which are then matched with a cartographic map based on road patterns (similar techniques have been used to localise noisy GPS data, see e.g. [23]).

The map matching approach is extended in [20] to include semantic features extracted from forward facing images - sun direction, road type, and junction presence - which are integrated with vehicle speed and odometry to estimate location and heading w.r.t the 2-D map using recursive inference. Classifiers for road types and junction presence are trained using image labels derived from OSM. The work has clear similarities with ours, but differs in that the semantic features derived from the map, i.e. road type and junctions, are used for street identification in order to constrain the map matching. This contrasts with our descriptor based

[1] www.openstreetmap.org

approach in which the presence of semantic features is used to encode specific locations. This likely reflects the difference in application; we are interested in localising slow moving pedestrians or robots as opposed to moving road vehicles and thus semantic descriptors directly related to map locations are a natural choice. They also align with using representations which better reflect human map reading. Comparing the two methods is therefore difficult, not least because the results presented in [20] were obtained using a front facing camera, whereas we require 360 degree views at each location.

III. OVERVIEW

The main components of the approach are illustrated in Figure 2. From a 2-D vector map, i.e. OSM, we generate binary semantic descriptors (BSDs) for locations spaced at regular intervals along roads in an urban environment. Each descriptor consists of 4 bits, with each bit indicating the presence or not of a semantic feature in a given viewing direction. For this initial investigation, we used only four directions - front, back, left and right facing - and two feature types - junctions and gaps between buildings. The latter were chosen since they are easily identified in the vector map and as described below, they can be reliably detected in images using trained classifiers. We experimented with alternative features such as building size and distance from roadways but found that gaps and junctions gave the best performance. In the future we plan to investigate using multiple directions and multiple features in each direction.

A database of location tagged route descriptors is then created by computing all possible routes within the area of interest up to a certain length in terms of the number of adjacent locations and then concatenating the set of associated BSDs as indicated in Figure 2d, where the circular discs represent the BSDs and the black/white segments indicate individual bits. Note that each route descriptor is then of length $4N_r$ bits, where N_r is the number of locations in the route. Thus, although the number of possible routes can be very large, the route database has a small memory footprint. For example, in the experiments described below, for an area of approximately 2.5 km^2, the number of possible routes containing 40 locations (each approximately 400 meters long represented by a 160-bit route descriptor) is just under 40×10^6. The route descriptor database is then around 800 MB in raw form, i.e. prior to any compression, which would be possible due to significant overlap between routes. This contrasts, for example, with the 177 GB reported in [5] required for image features to represent a single 1000 km route, i.e. equivalent to 71 MB for a single 400 meter route.

Localisation w.r.t the map then proceeds as follows. Images in the four viewing directions are captured at a location, i.e. within GSV in our case. Each image is then fed to a binary classifier, which detects the presence or not of a semantic feature, i.e. a junction for the front and back facing views and a gap between buildings for the left and right facing views. This gives a 4-bit BSD as illustrated in Figure 1, with each bit indicating the presence or not of the feature in each viewing direction.

The above BSD could be compared with those for all locations in the 2-D map to give localisation, but as noted earlier, their simplicity means that they are not sufficiently distinctive, with many locations having the same descriptor. Instead, as shown in in Figure 2a-c, we concatenate BSDs as the 'user' moves along a route in the environment, capturing images and generating descriptors at regular intervals, creating a route descriptor. In our case, we have a virtual user moving in GSV and generate BSDs at each successive GSV location (approximately every 10 meters). At each location, the current route descriptor is then used to query the database, with Hamming distances used to provide a ranked list of likely locations, as illustrated in Figure 2d-g.

To add further discrimination, we also compare the *turn patterns* - the position of left or right turns in a route - associated with the query and database routes, requiring that these are identical for a valid match. The motivation here is that direction changes of, for example, an autonomous vehicle can be detected reliably and hence can be used to eliminate spurious matches between route descriptors. The database route having the lowest Hamming distance w.r.t the query route and also the same turn pattern then provides the location estimate.

In the following sections, we provide details of the BSD generation, the design and training of the binary classifiers, the generation and comparison of the turn patterns and a probabilistic interpretation of the approach. Section VIII provides details of the GSV/OSM experiments and results and we conclude with a brief discussion of future work.

IV. BINARY SEMANTIC DESCRIPTORS

We denote the finite set of locations in an area of interest by $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$, where N is the total number of locations. Associated with each location l_i is a BSD, which we denote by the binary string d_i, with d_{ij} denoting the jth bit, and define $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ as the set of all descriptors. In this work, $j \in \{1, 2, 3, 4\}$ and each bit of a BSD denotes the presence or not of a junction or a gap between buildings in one of four viewing directions centred on location i. These are derived from the vector map as follows

$$d_{ij} = \begin{cases} JUNC(V_{ij}) & \text{if } j \in \{1, 2\} \\ BGAP(V_{ij}) & \text{if } j \in \{3, 4\} \end{cases} \quad (1)$$

where (V_{i1},V_{i2}) and (V_{i3},V_{i4}) denote the (front,back) and (left,right) viewing directions at location i, respectively. The functions $JUNC(V_{ij})$ and $BGAP(V_{ij})$ return 1 if there exists a junction or a gap between buildings, respectively, in direction V_{ij}, and 0 otherwise. As illustrated in Figure 3, a feature is deemed to be present in a viewing direction if one lies within the relevant quadrant of a circle of a given radius centred on the location of interest, where the front and back viewing directions are aligned with that of the road upon which the location sits. In the experiments we set the viewing distance radius to be 30m, which is similar to that used in [19].

For localisation we need to estimate a BSD for a location from images captured in each of the four viewing directions.

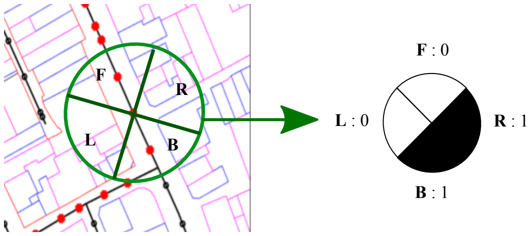


Fig. 3. Generation of a BSD from the vector map.

We do this using binary classifiers, trained to detect the presence or not of the relevant semantic feature. Given image I_{ij} at location i in viewing direction V_{ij}, the estimated BSD is given by

$$\hat{d}_{ij} = \begin{cases} DETECT_{JUNC}(I_{ij}) & \text{if } j \in \{1,2\} \\ DETECT_{BGAP}(I_{ij}) & \text{if } j \in \{3,4\} \end{cases} \tag{2}$$

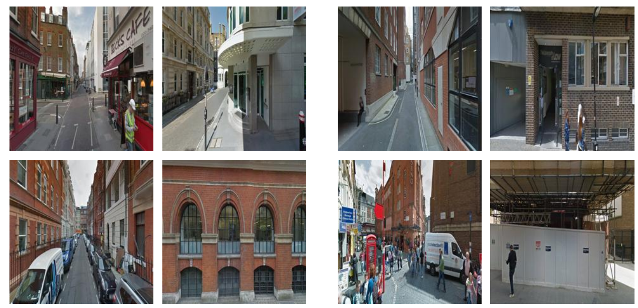
where $DETECT_{JUNC}(I_{ij})$ and $DETECT_{BGAP}(I_{ij})$ return 1 if a junction or a gap, respectively, are detected in image I_{ij}, and 0 otherwise, i.e. they mirror the BSD generation functions in Equation 1.

We use a CNN approach to design the binary classifiers $DETECT_{JUNC}$ and $DETECT_{BGAP}$. For training data, we make use of the correspondence between the vector maps in OSM and the images in GSV in a similar manner to that used in [19]. For each feature type - junctions and gaps between buildings - we collect positive samples by identifying the locations of the relevant features in OSM and storing the images from the corresponding locations (based on latitude and longitude) and relevant viewing directions from GSV, ensuring that we get a uniform mix of viewing scenarios. For example, in the case of junctions, we use front and back facing images aligned with the road and ensure that we have examples that cover the range of distances from the junction up to the viewing radius used in the generation of the BSDs. The training set is then completed by collecting approximately the same of number of negative samples in the corresponding viewing directions but not containing the feature of interest. In the experiments, we used a training sets consisting of 440,000 images per classifier taken from 220,000 locations in 23 different cities in the UK. None of these locations were used for evaluating the classifiers or in the localisation experiments.

We implemented the classifiers by using our training dataset to fine-tune an off-the-shelf pre-trained CNN. Specifically, we started from the pre-trained Places205-AlexNet model [24], designed for scene classification in urban environments, which aligns with our application, and derived from the pre-trained AlexNet model [25].We used colour images cropped from GSV panoramas in the required viewing direction corresponding to a $90°$ horizontal field of view and resized to 227×227 pixels. The latter results in some distortion but given that we used the same process for both training and testing, this was not deemed to be an issue. Examples of positive and negative images from the training dataset are shown in Figure 4. We tested performance of each classifier using two test sets of 8000 images taken from the same 23 cities but at locations not within the training set and with an equal number of positives and negatives samples, i.e. feature present and not present.

(a) (b)

Fig. 4. Examples of positive (feature present) and negative (feature not present) images from the training datasets used for the semantic classifiers: (a) junction (top) and no junction (bottom); (b) gap (top) and no gap (bottom).



(a) (b)

Fig. 5. Examples of semantic classifications: (a) true positives (top) and true negatives (bottom); (b) false positives (top) and false negatives (bottom). In both (a) and (b) examples are arranged as: junction (top-left); gap (top-right); no junction (bottom-left); no gap (bottom-right).

Both classifiers gave good balanced performance in detecting the presence and non-presence of junctions and gaps, with precision and recall values of 0.75 ± 0.02 on the test set. Examples of correct classifications (true positives and true negatives) and incorrect classifications (false positives and false negatives) are shown in Figure 5. Note that the latter illustrate the difficulty of the task. For example, the bottom left view in Figure 5b contains a junction which is significantly obscured and was incorrectly classified as containing no junction, whilst the 2-D map indicates that the bottom right view should contain a gap, but the site appears to be under redevelopment and has been incorrectly classified as not containing a gap. The latter is an example of inaccuracies within the OSM data.

V. ROUTE DESCRIPTORS AND TURN PATTERNS

As noted earlier and as we demonstrate later, on their own the above binary descriptors are not sufficiently discriminative to identify a location uniquely and allow localisation. This is true even if we were able to design perfect classifiers for extracting the descriptors from images. The simplicity of the representation, whilst being extremely compact, means that there are many locations with similar descriptors. We address this ambiguity in two ways. First, we concatenate descriptors along routes corresponding to adjacent locations, constructing *route descriptors*, which prove to be highly discriminative once the routes reach a certain length. Once this length is reached, then localisation can proceed at the rate that new locations are visited, i.e. enabling tracking, by matching with a database of all possible route descriptors constructed offline. Secondly, we introduce further disambiguation by incorporating *turn patterns* observed along routes into the representation, i.e. the sequence no turn and turn (left or right) at each location along a route, and using these to identify the most likely match within the database.

Let A be an $N \times N$ adjacency matrix, such that $A_{ij} = 1$ if locations l_i and l_j are adjacent, and $A_{ij} = 0$ otherwise. Locations are regarded as adjacent if on the 2-D map they are connected by a road and there are no other locations between them. A *route* is then defined as a finite sequence of adjacent locations, i.e. the route $r = (l_{\gamma(1)}, l_{\gamma(2)}, \ldots, l_{\gamma(N_r)})$ is of length N_r, where $\gamma(i)$ defines a sequence of adjacent locations such that $A_{\gamma(i)\gamma(i+1)} = 1$, $\forall\ 1 \leq i < N_r$. For simplicity we have restricted ourselves to routes that do not loop or turn back on themselves, i.e. $\gamma(i) \neq \gamma(j)$, $\forall\ i \neq j$, $1 \leq i, j \leq N_r$, but the method could be readily extended to deal with such cases. We define $\mathcal{R}_M$ as the set of all such routes up to length M defined amongst all the locations in $\mathcal{L}$. Associated with each route is a *route descriptor*, consisting of the sequence of BSDs corresponding to the locations along the route, i.e. $s = (d_{\gamma(1)}, d_{\gamma(2)}, \ldots, d_{\gamma(N_r)})$, and we define $\mathcal{S}_M$ as the set of all route descriptors corresponding to the routes in $\mathcal{R}_M$.

To incorporate turn information into the representation, we define a binary *turn pattern* $t = (t_{\gamma(1)}, t_{\gamma(2)}, \ldots, t_{\gamma(N_r-1)})$ associated with a route r. The ith bit of t indicates whether a left and right turn is present between locations $l_{\gamma(i)}$ and $l_{\gamma(i+1)}$, i.e. $t_{\gamma(i)} = TURN(\theta_{\gamma(i)}, \theta_{\gamma(i+1)})$, where $\theta_{\gamma(i)}$ denotes the front facing direction at location $l_{\gamma(i)}$ and

$$TURN(\theta_i, \theta_j) = \begin{cases} 1 & \text{if } \lfloor \theta_i - \theta_j \rceil \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\lfloor \theta_i - \theta_j \rceil$ denotes the absolute value of the smallest angle between θ_i and θ_j, and τ is an angle threshold, which we set to be 60° to ensure that we only include significant turns. Thus t represents the sequence of turns that take place along a route. We define $\mathcal{T}_M$ to be the set of such turn patterns corresponding to the routes in $\mathcal{R}_M$.

VI. LOCALISATION AND BOOTSTRAPPING

Consider an autonomous system making its way through an urban environment, moving between locations in $\mathcal{L}$ along a specific route of length $< M$. At any given location, our goal is to identify its current location by recognising the route taken to date, consisting of the current location plus the previous $N_r - 1$ locations, say. We do this by comparing its estimated route descriptor $\hat{s}$ (obtained by concatenating the estimated BSDs at each location) with those in $\mathcal{S}_{N_r} \subset \mathcal{S}_M$ and its turn pattern $\hat{t}$ with those in $\mathcal{T}_{N_r} \subset \mathcal{T}_M$, hence determining the most likely route from those in $\mathcal{R}_{N_r} \subset \mathcal{R}_M$.

It is important to note that in this work we assume that there is a one-to-one correspondence between the locations in our 2-D map and the locations in the environment. This enables us to do a direct comparison between estimated route descriptors and those in the database. When using GSV and OSM data this can be ensured by selecting OSM locations corresponding to the known locations in GSV. In a practical system, we would need a method of forming such one-to-one correspondence or alternatively, a means overcoming the lack of it.

We define the most likely route $r^* \in \mathcal{R}_{N_r}$ as being the route whose route descriptor s^* is closest to $\hat{s}$ and whose turn pattern t^* matches $\hat{t}$, i.e. such that

$$s^* = \underset{s \in \mathcal{S}_{N_r}}{\arg\min}\, DIST(s, \hat{s}) \tag{4}$$

and

$$DIST(t^*, \hat{t}) = 0 \tag{5}$$

where $DIST(a, b)$ denotes the Hamming distance between two binary strings a and b. For long routes ($N_r > 20$, say) the number of elements in $\mathcal{S}_{N_r}$ becomes very large ($> 500 \times 10^3$, rising to near 40×10^6 for $N_r = 40$), and thus we use an efficient pattern matching algorithm based on a BK-tree [26] to find the closest route descriptor in Equation (4).

Note from the above that we assume that the turn pattern for the query route is correct, but allow errors in the estimate of the route descriptor due to the non-perfect classifiers used in the semantic feature detectors. Our motivation for the former is that in practice detecting significant left or right turns by an autonomous system can be achieved reliably and thus requiring an exact match is reasonable. Note, however, that as we show later, turn patterns alone are not sufficient to achieve localisation, as many routes share the same turn pattern, and it is their combination with route descriptors that gives the required level of distinctiveness.

The above provides an indication of the most likely location given the current route. However it gives no indication as to the confidence in the estimate. There are a number of possibilities for this, including basing it on the distance between s^* and $\hat{s}$ and/or the distance of s^* from the second best matching route descriptor. We found that a consistency metric proved to be most effective, in which we deem a route to be localised if there is sufficient overlap between the most likely routes r^* for a number of successive locations. We set the overlap to be 80% of the locations need to be the same and we required this to occur for 5 successive locations. In essence, if successive query routes are matching with routes that have significant overlap then it is a good indicator that successful localisation has been achieved.

We also demonstrate later that once the above consistency criterion has been met, the query route length can be fixed and localisation proceed by successively updating the query route by appending the latest BSD onto the end and removing the first descriptor. Thus, the phase during which the query route grows can be regarded as a *bootstrapping process*, during which the route descriptor extends until it becomes sufficiently distinct to allow localisation. Once complete, then *continuous tracking* can take place using the fixed length query at the same rate as the BSD are created at successive locations. An example of bootstrapping and tracking is shown in the video submitted as supplementary material.

VII. PROBABILISTIC FORMULATION

The above localisation process can also be considered in probabilistic terms. Given an estimated BSD $\hat{d}$ obtained at a single location l, say, then the conditional probability that l corresponds to $l_i \in \mathcal{L}$ can be written as

$$P(l_i|\hat{d}) = P(l_i|d_i)P(d_i|\hat{d}) \propto P(l_i|d_i)P(\hat{d}|d_i) \qquad (6)$$

where we assume that all descriptors d_i are equally likely. Note that the term $P(l_i|d_i)$ expresses the uniqueness of the ground-truth descriptor d_i derived from the 2-D map. Since our descriptors are only 4-bits long, then for a large number of locations, e.g. 6000 in the experiments, $P(l_i|d_i) << 1$, indicating that many locations have the same descriptors and hence localisation is not possible. Given that we have an estimate of the accuracy of our classifiers and hence the detectors $DETECT_{JUNC}$ and $DETECT_{BGAP}$, we can approximate the likelihood $P(\hat{d}|d_i)$ in terms of the Hamming distance h between d_i and $\hat{d}$, i.e.

$$P(\hat{d}|d_i) \propto q^{4-h}(1-q)^h \qquad (7)$$

where q is the probability of correctly detecting the presence or not of both junctions and gaps (we assume the same value for both probabilities for simplicity, but as noted in Section IV, we also observed similar values in practice of ≈ 0.75).

Extending the above to routes, we obtain the following conditional probability that the route descriptor estimate $\hat{s} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{N_r})$ corresponds to route $r \in \mathcal{R}_{N_r}$

$$P(r|\hat{s}) = P(r|s)P(s|\hat{s}) = P(r|s)P(\hat{s}|s) \qquad (8)$$

Hence from Equations (6) and (7) and assuming independence between descriptors

$$P(r|\hat{s}) \propto P(r|s)\prod_{i=1}^{N_r} P(\hat{d}_i|d_{\gamma(i)}) \qquad (9)$$

$$\propto P(r|s)q^{4N_r-H}(1-q)^H \qquad (10)$$

where H denotes the Hamming distance between s and $\hat{s}$. Here, $P(r|s)$ expresses the uniqueness of the route descriptor s, which as we demonstrate later is high for a sufficiently long routes and thus $P(r|s) \to 1$, giving

$$P(r|\hat{s}) \propto q^{4N_r-H}(1-q)^H \qquad (11)$$

Using this expression we can obtain an estimate of the likelihood ratio of one route r_i over another r_j for a given $\hat{s}$

$$\frac{P(r_i|\hat{s})}{P(r_j|\hat{s})} = \frac{(1-q)^{H_i-H_j}}{q^{H_i-H_j}} \qquad (12)$$

where H_i is the Hamming distance between s_i and $\hat{s}$. Hence for $q = 0.75$, this gives a likelihood ratio of $1/3^\delta$ for a difference δ in Hamming distance from the estimated route descriptor, which is significant. For example, a route whose descriptor is δ bits closer in Hamming distance to the estimated descriptor, is 3^δ times more likely to be the correct route. Thus, as we demonstrate in the next section, even with a detector accuracy of only 75% for individual BSDs, the concatenation of descriptors along routes can lead to a high degree of distinctiveness for long enough routes.

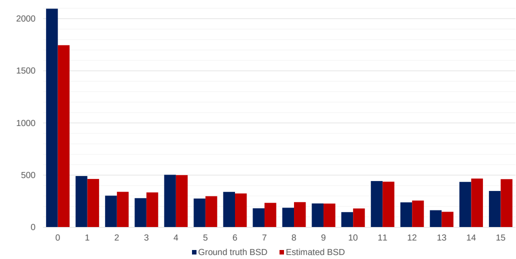


Fig. 6. Histogram showing the distribution of 4-bit ground-truth (blue) and estimated (red) BSDs obtained from OSM and GSV images, respectively.

VIII. EXPERIMENTS

We evaluated the performance of the method using GSV and OSM data for a 2.5 km^2 region of London. None of the locations within the region were used to train the semantic classifiers. The region consisted of 6656 GSV locations and from each location we gathered images corresponding to the four viewing directions, from which we estimated 4-bit BSDs using the classifiers described in Section IV.

To illustrate the distribution of descriptors across the region and the performance of the classifiers, Figure 6 shows the histogram of 4-bit ground-truth descriptors (obtained from OSM, shown in blue) and estimated descriptors, shown in red, where the horizontal axis corresponds to the 16 possible 4-bit patterns. The predominance of BSDs with pattern '0000' corresponding to locations in which there is neither gaps between buildings to the left or right, nor junctions towards the front or back, results from the fact that many locations between junctions have these characteristics as can be seen from the 2-D map of the area shown in Figure 11. Note that the distribution of the estimated descriptors is close to that of the ground-truth due to the accuracy of the classifiers, i.e. approximately 75%.

To assess the performance of the route based localisation, we considered route lengths up to a maximum of $M = 40$ locations. We tested the method using 150 test routes and for each we recorded the route length at which localisation was achieved according to the consistency criterion, i.e. 5 successive consistent localisations. The results are shown in Figure 7, which shows the percentage of routes that were correctly localised within route lengths of 0-5, 0-10, ..., 0-40 locations. We have shown the results for three methods of matching routes: using only turn patterns (grey); using only route BSDs (yellow); and using both BSDs and turn patterns (blue). Note that the latter outperforms the others by a significant margin and that BSDs alone also significantly outperform turn patterns, which only manage to localise $<$ 10% of routes even with a route length of 40 locations. This clearly demonstrates the potential of the BSD approach. Note in particular that over 85% of routes are correctly localised even when only using routes consisting of up to 20 locations, which corresponds to approximately 200 meters in GSV.

It is also interesting to consider the significance of the classifier accuracy. Given that we know the ground-truth BSDs, we investigated using BSDs 'estimated' using classifiers with different accuracy (we assumed the same accuracy

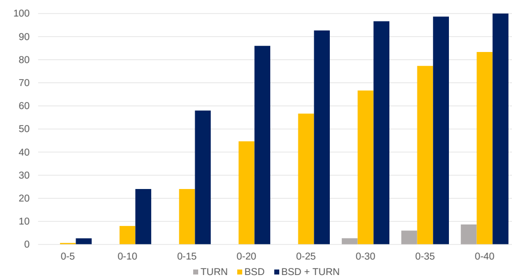


Fig. 7. Accuracy of localisation (% of correctly identified routes) versus route length using turn patterns (grey), route descriptors (yellow), and route descriptors with turn patterns (blue).

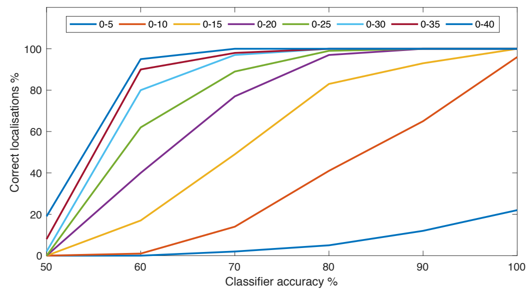


Fig. 8. Accuracy of localisation (% of correctly identified routes) versus classifier accuracy for different ranges of route length.

for both detecting the presence or not of junction and gaps). A plot of the percentage of correctly localised routes versus the accuracy of the classifiers is shown in Figure 8 for the different route length ranges used in Figure 7. Thus that at 75% accuracy, which we obtained from our trained classifiers, over 85% of routes are predicted to be correctly localised within 0-20 locations, which agrees with our findings in Figure 7. Note also that if classifier accuracy were increased to beyond 80%, then 80-90% of routes could be correctly localised using < 15 locations, which again illustrates the potential of the BSD approach.

Examples of estimated BSDs, their corresponding images in the four viewing directions and the ground-truth BSDs from OSM for part of a route are shown in Figure 9. Note the deviation of the BSD estimates from the ground-truth, which results from the inaccuracy of the classifiers. The challenging nature of the detection task can be seen from the images. This confirms the utility of concatenating BSDs along a route in order to gain uniqueness and hence enable localisation. Figure 10 further illustrates this, which shows the distribution of Hamming distances from descriptors in the database for a given test (query) route at lengths of 15 (left) and 30 (right) locations, with and without using turn patterns (bottom and top, respectively). The correct matches for lengths 15 and 30 have Hamming distances of 15 and 26, respectively. When the test route length is 15 locations, the correct route is not the closest (there are other Hamming distances with values < 15), although using turns (bottom) significantly reduces the number of routes close to the query route (note that the vertical axes in Figure 10 have significantly different ranges). With 30 locations and without using turns, the correct route does become equal closest with 18 others and there are a significant number of others close by. In contrast, using turn

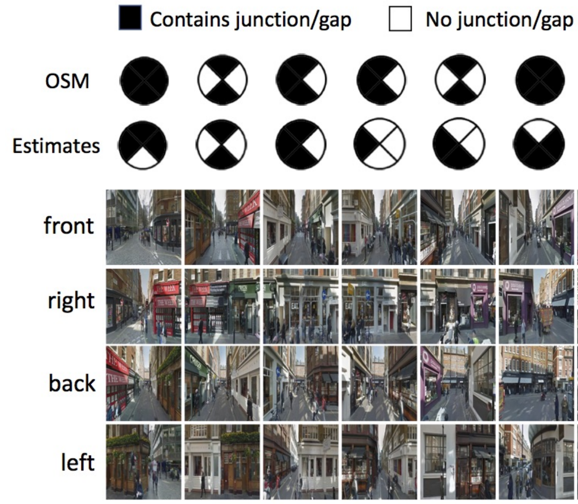


Fig. 9. Examples of BSD ground-truths, from OSM, and BSD estimates, from the classification of the GSV images in four directions as shown.

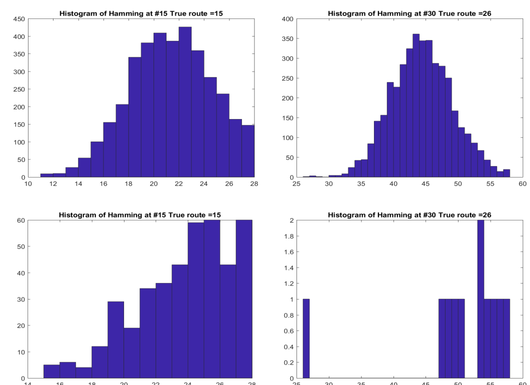


Fig. 10. Histograms of Hamming distances between a test route descriptor and those in the database for route lengths of 15 (left) and 30 (right) locations, with (bottom) and without (top) using turn patterns.

patterns with 30 locations drastically reduces the number of candidate routes and the correct route becomes the closest by a Hamming distance margin of over 20.

To illustrate the localisation process, Figure 11 shows snapshots of the localisation of a test route at route lengths of 2 and 24 locations. It shows the OSM 2-D map and the locations are indicated by coloured squares along roads, where the colour indicates the closeness between their route descriptor and that of the test route (where locations share routes, then the closest descriptor difference is shown). The latest location along the test route is indicated by the orange/red circle, where orange indicates that the route has yet to be correctly and consistently localised, and red indicates that localisation has been achieved. The BSDs, estimated and ground-truths, along with their images, are shown below the 2-D maps. Note that the bottom row of images show the views at the closest (best) match locations, but are not used in the matching process. With route length of 2, the majority of locations have a low likelihood of being correct (dark blue), whilst a small number of disparate locations have a high likelihood (dark red). This reflects the lack of distinctiveness of two 4-bit BSDs. In contrast, once 24 locations are reached, the route has been successfully

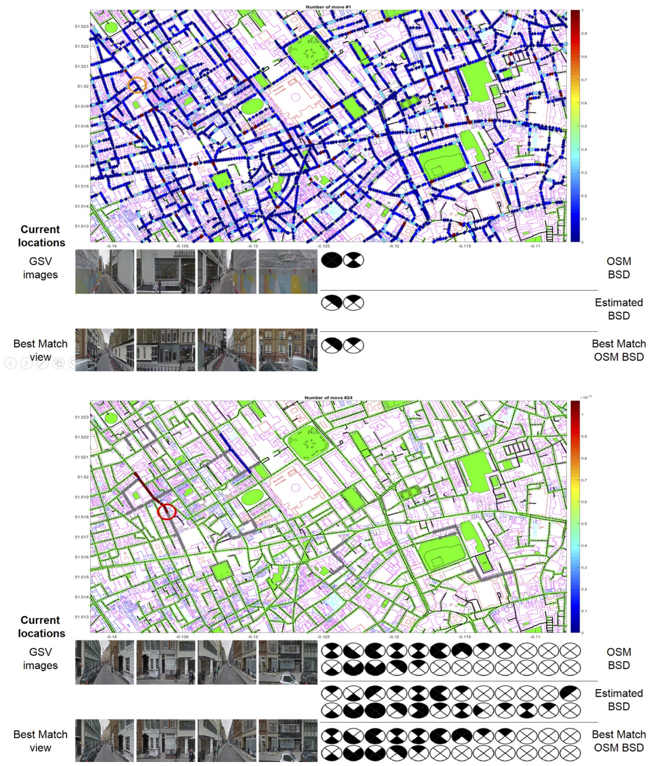


Fig. 11. Snapshots of the localisation process for test route lengths of 2 (top) and 24 locations (bottom). See text for explanation.

localised and the vast majority of other locations/routes have been eliminated (their squares are not shown), reflecting the confidence of the localisation. A video showing the complete process has been submitted as supplementary material.

IX. CONCLUSIONS

We have presented a novel approach to position localisation in urban areas and to the best of our knowledge it is the first example of linking 2-D maps to images over large areas. The key contribution is the demonstration that compact binary semantic descriptors concatenated over routes are sufficiently distinctive to enable localisation and that the representation is vastly smaller than that used in image to image database approaches. Moreover, the use of simple semantic classification offers the potential for invariance to changing environment conditions, which is something that we wish to demonstrate in future. In addition, the reported work relies on an assumption of one-to-one correspondence between map and image locations, achieved by using OSM and GSV data, and this needs to be addressed for developing a practical system, which we are in the process of doing.

REFERENCES

[1] S. Lowry, N. Sunderhauf, P. Newman, J. Leonard, D. Cox, P. Corke, and M. Milford, "Visual place recognition: A survey," *IEEE Trans. on Robotics*, vol. 32, pp. 1–19, 2016.

[2] K. Lynch, *The image of the city*. MIT Press, 1960.

[3] J. O'Keefe and L. Nadel, *The hippocampus as a cognitive map*. Clarendon Press, 1978.

[4] B. Tversky, "Cognitive maps, cognitive collages, and spatial mental models," in *Proc. Euro. Conf. on Spatial Info. Theory (COSIT)*, 1993.

[5] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. of Robotics Research*, vol. 30, pp. 1100–1123, 2010.

[6] M. Milford and G. Wyeth, "Seqslam : visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Int. Conf. on Robotics and Automation*, 2012.

[7] P. Biber and T. Duckett, "Experimental analysis of sample-based maps for long-term slam," *Int. J. of Robotics Research*, vol. 28, no. 1, 2009.

[8] S. Lowry, M. Milford, and G. Wyeth, "Transforming morning to afternoon using linear regression techniques," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014.

[9] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proc. of Robotics: Science and Systems*, 2015.

[10] P. Panphattarasap and A. Calway, "Visual place recognition using landmark distribution descriptors," in *Proc Asian Conf. on Computer Vision*, 2016.

[11] R. Frampton and A. Calway, "Place recognition from disparate views," in *Proc. British Machine Vision Conf. (BMVC)*, 2013.

[12] A. Mousavian and J. Kosecka, "Semantically guided location recognition for outdoor scenes," in *Proc IEEE Int. Conf. on Robotics and Automation*, 2015.

[13] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Localization from semantic observations via the matrix permanent," *Int J of Robotics Research*, vol. 35, no. 1-3, pp. 73–99, 2016.

[14] T.-J. Cham, A. Ciptadi, W.-C. Tan, M.-T. Pham, and L.-T. Chia, "Estimating camera pose from a single urban ground-view omnidirectional image and a 2d building outline map," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[15] A. Mousavian and J. Kosecka, "Semantic image based geolocation given a map," *ArXiv e-prints*, 2016.

[16] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit, "Instant outdoor localization and slam initialization from 2.5d maps," *IEEE Trans. on Vis. and Computer Graphics*, vol. 21, no. 11, 2015.

[17] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit, "Accurate camera registration in urban environments using high-level feature matching," in *Proc. British Machine Vision Conf.*, 2017.

[18] A. Armagan, M. Hirzer, P. Roth, and V. Lepetit, "Learning to align semantic segmentation and 2.5 d maps for geolocalization," in *IEEE Conf. on Computer Vision and Pattern Recognitiion*, 2017.

[19] A. Seff and J. Xiao, "Learning from maps: Visual common sense for autonomous driving," *ArXiv e-prints*, 2016.

[20] W.-C. Ma, S. Wang, M. A. Brubaker, S. Fidler, and R. Urtasun, "Find your way by observing the sun and other semantic cues," in *IEEE Int Conf on Robotics and Automation*, 2017.

[21] G. Floros, B. van der Zander, and B. Leibe, "Openstreetslam: Global vehicle localization using openstreetmaps." in *IEEE Int Conf on Robotics and Automation*, 2013.

[22] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based Probabilistic Visual Self-Localization," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 652 – 665, 2016.

[23] A. Javanmard, M. Haridasan, and L. Zhang, "Multi-track map matching," in *Proc Int Conf on Advances in Geographic Information Systems*, 2012.

[24] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Int. Conf. on Neural Information Processing Systems*, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. on Neural Information Processing Systems*, 2012.

[26] W. A. Burkhard and R. M. Keller, "Some approaches to best-match file searching," *Commun. ACM*, vol. 16, no. 4, 1973.